UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/849,025	05/20/2004	Tal Gat	P-6618-US	7037

49444 7590 11/02/2006

PEARL COHEN ZEDEK LATZER, LLP
1500 BROADWAY, 12TH FLOOR
NEW YORK, NY 10036

EXAMINER	
LAI, VINCENT	
ART UNIT	**PAPER NUMBER**
2181	

DATE MAILED: 11/02/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/849,025	**Applicant(s)** GAT, TAL	
	Examiner Vincent Lai	**Art Unit** 2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 August 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

10/29/2006

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. Acknowledgment is made of the amendments to the specification, title, abstract, and claims.

Response to Arguments

2. Applicant's arguments filed 2 August 2006 have been fully considered but they are not persuasive.

Arguments are addressed below in the rejection of the claims. Examiner does not believe claims have been changed since emphasis of an "array able to maintain a second number of entries smaller than said first number of entries" was already claimed and was argued by Examiner. Applicant did not make an argument on rejection but merely stated disagreement.

Examiner also notes that claim 18 was not properly marked as being currently amended and is assumed to be.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States

only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1, 3-6, 8-9, 11-13, 16-17, and 19-21 are rejected under 35 U.S.C. 102(e) as being anticipated by Arora et al (U.S. Patent # 6,629,238 B1), herein referred to as Arora.

As per **claim 1**, Arora discloses a loop detector (See column 3, lines 63-65: The invention deals with a loop predictor which can be interpreted as a loop detector) comprising:

a set of branch entries to maintain data relating to a set of branches, respectively, wherein said set of branch entries includes a first number of entries (See figure 8 and column 11, lines 7-10: A program segment 824 (or 854 as shown in the figure) can be made up of loop instructions); and

an array able to maintain a set of iteration entries (See figure 8 and column 11, lines 7-10: A memory 820 can be used to store loops), wherein the set of iteration entries includes a second number of entries smaller than said first number of entries (See figure 8: Pictorially, the program segment is smaller than the memory. Also since the program segment is meant to fit in memory, it is inherently smaller than memory);

wherein said loop detector is able to allocate at least one of said iteration entries to store loop iteration data relating to at least one branch entry of said set of branch entries, respectively (See column 11, lines 7-10: Saving to memory means that loop detector will store data)

As per **claim 3**, Arora discloses wherein the loop iteration data relating to said at least one branch entry comprises at least one counter to count speculative iterations of said at least one branch (See column 4, lines 36-38, and column 12, claim 12: Other loop counters are disclosed/claimed including a speculative epilog counter).

As per **claim 4**, Arora discloses wherein the loop iteration data relating to said at least one branch entry comprises at least one counter to count real iterations of said at least one branch (See column 4, lines 36-38: Epilog counters are used to count completed loops).

As per **claim 5**, Arora discloses a method of storing a counter of loop iterations, the method comprising:

maintaining data relating to a set of branches in a set of respective branch entries of a loop detector (See column 5, lines 17-18: An example of data maintenance is shown; loop counters are used), wherein said set of branch entries includes a first number of entries (See figure 8: A memory 820 can be used to store loops);

determining if loop iteration data for a branch of said set of branches is stored in an entry of an array associated with said loop detector, said array able to maintain a second number of entries smaller than the first number of entries (See column 5, lines 17-18: If loop iteration is already stored then data would be available).

incrementing a counter in said entry (See column 5, lines 17-18 and column 7, lines 57-60: Loop counters are initialized though moving loop instructions and that value is stored); wherein the number of entries in said array is smaller than the number of entries in said loop detector.

As per **claim 6**, Arora discloses comprising copying a number of actual iterations of said branch into said entry (See column 7, lines 57-60: The MOV_TO_LC instruction moves loop instructions into the array).

As per **claim 8**, Arora discloses wherein incrementing said counter comprises incrementing a counter of actual iterations of said branch (See column 4, lines 36-38: Epilog counters are used to count completed loops).

As per **claim 9**, Arora discloses wherein incrementing said counter comprises incrementing a counter of speculative iterations of said branch (See column 4, lines 36-38, and column 12, claim 12: Other loop counters are disclosed/claimed including a speculative epilog counter).

As per **claim 11**, Arora discloses a method of counting loop iterations, comprising:

maintaining data relating to a set of branches in a set of respective branch entries of a loop detector (See column 5, lines 17-18: An example of data maintenance is shown; loop counters are used), wherein said set of branch entries includes a first number of entries (See figure 8: A memory 820 can be used to store loops); and allocating an entry of an array associated with said loop detector to store loop iteration data relating to a branch entry of said branch entries (See column 11, lines 7-10: Saving to memory means that loop detector will store data),

wherein said array is able to maintain a set of iteration entries (See figure 8 and column 11, lines 7-10: A memory 820 can be used to store loops), said set of iteration entries includes a second number of entries smaller than said first number of entries (See figure 8: Pictorially, the program segment is smaller than the memory. Also since the program segment is meant to fit in memory, it is inherently smaller than memory).

As per **claim 12**, Arora discloses comprising storing is said entry a counter of speculative loop iterations of said branch (See column 4, lines 36-38, and column 12, claim 12: Other loop counters are disclosed/claimed including a speculative epilog counter).

As per **claim 13**, Arora comprising storing in a said entry a counter of real loop iterations of said branch (See column 4, lines 36-38: Epilog counters are used to count completed loops).

As per **claim 16**, Arora discloses a processor comprising a loop detector, said loop detector comprising:

a set of branch entries to maintain data relating to a set of branches, respectively, wherein said set of branch entries includes a first number of entries (See figure 8 and column 11, lines 7-10: A program segment 824 (or 854 as shown in the figure) can be made up of loop instructions); and

an array able to maintain a set of iteration entries (See figure 8 and column 11, lines 7-10: A memory 820 can be used to store loops), wherein the set of iteration entries includes a second number of entries smaller than said first number of entries (See figure 8: Pictorially, the program segment is smaller than the memory. Also since the program segment is meant to fit in memory, it is inherently smaller than memory);

wherein said loop detector is able to allocate at least one of said iteration entries to store loop iteration data relating to at least one branch entry of said set of branch entries, respectively (See column 11, lines 7-10: Saving to memory means that loop detector will store data).

As per **claim 17**, Arora discloses wherein said loop iteration data comprises a speculative counter of loop iterations (See column 4, lines 36-38, and column 12, claim 12: Other loop counters are disclosed/claimed including a speculative epilog counter).

As per **claim 19**, Arora discloses a system comprising:

a dynamic random access memory unit (Memory 820, see figure 8); and

a processor comprising a loop detector (See column 3, lines 63-65: The invention deals with a loop predictor which can be interpreted as a loop detector), said loop detector comprising:

a set of branch entries to maintain data relating to a set of branches, respectively, wherein said set of branch entries includes a first number of entries (See figure 8 and column 11, lines 7-10: A program segment 824 (or 854 as shown in the figure) can be made up of loop instructions); and

an array able to maintain a set of iteration entries (See figure 8 and column 11, lines 7-10: A memory 820 can be used to store loops), wherein the set of iteration entries includes a second number of entries smaller than said first number of entries (See figure 8: Pictorially, the program segment is smaller than the memory. Also since the program segment is meant to fit in memory, it is inherently smaller than memory);

wherein said loop detector is able to allocate at least one of said iteration entries to store loop iteration data relating to at least one branch entry of said set of branch entries, respectively (See column 11, lines 7-10: Saving to memory means that loop detector will store data).

As per **claim 20**, Arora discloses wherein said loop iteration data comprises a counter to count speculative iterations of said branch (See column 4, lines 36-38, and column 12, claim 12: Other loop counters are disclosed/claimed including a speculative epilog counter).

As per **claim 21**, Arora discloses wherein said loop iteration data comprises a counter to count real iterations of said branch (See column 4, lines 36-38: Epilog counters are used to count completed loops).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 2, 10, 15, and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Arora et al (U.S. Patent # 6,629,238 B1), herein referred to as Arora in view of the Intel Itanium Processor Microarchitecture Reference.

As per **claim 2**, Arora teaches the use of the Intel IA-64 architecture (See column 4, lines 27-30).

Arora does not teach wherein entries in said array are fully associative.

The Intel Itanium Processor Microarchitecture References, which teaches an implementation of the Intel IA-64 architecture (See page 1, section 1.0 Overview), teaches the use of fully associative cache (See page 14, section 4.7 Translation Lookaside Buffers)

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Arora to include fully associative cache because the disclosed architecture already supports fully associative cache.

As per **claim 10**, Arora teaches the use of the Intel IA-64 architecture (See column 4, lines 27-30).

Arora does not teach wherein entries in said array are fully associative.

The Intel Itanium Processor Microarchitecture References, which teaches an implementation of the Intel IA-64 architecture (See page 1, section 1.0 Overview), teaches the use of fully associative cache (See page 14, section 4.7 Translation Lookaside Buffers)

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Arora to include fully associative cache because the disclosed architecture already supports fully associative cache.

As per **claim 15**, Arora teaches allocating an entry (See column 5, lines 32-39: Branch loops are supported and they can be treated like normal loops, which includes placing them into an entry).

Arora does not teach wherein entries in said array are fully associative.

The Intel Itanium Processor Microarchitecture References, which teaches an implementation of the Intel IA-64 architecture (See page 1, section 1.0 Overview), teaches the use of fully associative cache (See page 14, section 4.7 Translation Lookaside Buffers)

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Arora to include fully associative cache because the disclosed architecture already supports fully associative cache.

As per **claim 18**, Arora teaches the use of the Intel IA-64 architecture (See column 4, lines 27-30).

Arora does not teach wherein entries in said array are fully associative.

The Intel Itanium Processor Microarchitecture References, which teaches an implementation of the Intel IA-64 architecture (See page 1, section 1.0 Overview), teaches the use of fully associative cache (See page 14, section 4.7 Translation Lookaside Buffers)

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Arora to include fully associative cache because the disclosed architecture already supports fully associative cache.

5. Claims 7 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Arora et al (U.S. Patent # 6,629,238 B1), herein referred to as Arora in view of the Inside the Intel Itanium 2 Processor.

As per **claim 7**, Arora teaches the use of the Intel IA-64 architecture (See column 4, lines 27-30).

Arora does not teach using the method of allocating an entry of said array based on the least recently used entry in said array.

The Inside the Intel Itanium 2 Processor, which is teaches implementation of the Intel IA-64 architecture (See page 4, point 2 of the Customer Benefits From Industry

Standard Processors section), teaches the use of a least recently used cache replacement algorithm (See page 30).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Arora to include the use of a least recently used cache replacement algorithm because the disclosed architecture already supports such cache replacement scheme.

As per **claim 14**, Arora teaches allocating an entry (See column 5, lines 32-39: Branch loops are supported and they can be treated like normal loops, which includes placing them into an entry) in said array and the use of the Intel IA-64 architecture (See column 4, lines 27-30).

Arora does not teach using the method of allocating an entry of said array based on the least recently used entry in said array.

The Inside the Intel Itanium 2 Processor, which is teaches implementation of the Intel IA-64 architecture (See page 4, point 2 of the Customer Benefits From Industry Standard Processors section), teaches the use of a least recently used cache replacement algorithm (See page 30).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Arora to include the use of a least recently used cache replacement algorithm because the disclosed architecture already supports such cache replacement scheme.

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Vincent Lai whose telephone number is (571) 272-6749. The examiner can normally be reached on M-F 8:00-5:30 (First BiWeek Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Fritz M. Fleming can be reached on (571) 272-4145. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Vincent Lai
Examiner
Art Unit 2181

vl
October 27, 2006

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100
10/29/2006